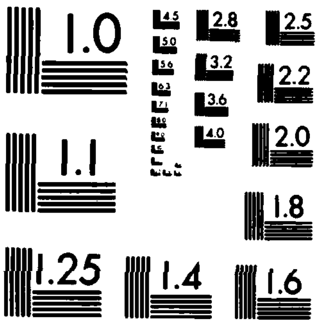
AD-A155 396   SPEECH COMPRESSION AND SYNTHESIS(U) BOLT BERANEK AND   1/1
NEWMAN INC CAMBRIDGE MA M BEROUTI ET AL. OCT 79
BBN-4266 F19628-78-C-0136

UNCLASSIFIED   F/G 9/4   NL

Re



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

Bolt Beranek and Newman Inc.

β

(1)

REPORT NO. 4266

SPEECH COMPRESSION AND SYNTHESIS

QUARTERLY PROGRESS REPORT NO. 5
8 JUNE 1979 - 7 SEPTEMBER 1979

PREPARED FOR:
ADVANCED RESEARCH PROJECTS AGENCY

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER
BBN Report No. 4266

2. GOVT ACCESSION NO.
AD-A155 396

3. RECIPIENT'S CATALOG NUMBER

4. TITLE (and Subtitle)
SPEECH COMPRESSION AND SYNTHESIS

5. TYPE OF REPORT & PERIOD COVERED
Quarterly Tech. Report
8 June 1979-7 Sept.1979

6. PERFORMING ORG. REPORT NUMBER

7. AUTHOR(s)
Michael Berouti
John Klovstad
John Makhoul
Richard Schwartz
John Sorensen

8. CONTRACT OR GRANT NUMBER(s)
F19628-78-C-0136

9. PERFORMING ORGANIZATION NAME AND ADDRESS
Bolt Beranek and Newman Inc.
50 Moulton St.
Cambridge, Massachusetts

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS

11. CONTROLLING OFFICE NAME AND ADDRESS

12. REPORT DATE
Ocotber 1979

13. NUMBER OF PAGES

14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)
Deputy for Electronic Technology (RADC/ETC)
Hanscom Air Force Base, MA 01731
Contract Monitor: Mr. Caldwell P. Smith

15. SECURITY CLASS. (of this report)
Unclassified

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)
Distribution of this document is unlimited. It may be released to the Clearinghouse, Department of Commerce, for sale to the general public.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES
This research was supported by the Defense Advanced Research Projects Agency under ARPA Order No. 3515.

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)
Speech synthesis, phonetic synthesis, diphone, LPC synthesis, vocoder, speech compression, linear prediction, voice-excited coder, high-frequency regeneration, spectral duplication, phonetic vocoder, spectral template, speech recognition, Phoneme recognition,

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)
This document reports on work towards a very low rate phonetic vocoder, text to speech, and multirate speech compression. This work included improvement of the phonetic synthesis algorithms and continued gathering of the diphone templates data base for phonetic synthesis. It also included the initial design of a phonetic recognizer to operate in conjunction with the synthesizer. The combination of these two programs will

Unclassified
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

result in the very low rate vocoder. The method of spectral harmonic deviation was tested in an LPC vocoder environment. The sources for the MIT text-to-speech system were translated from BCL to BCPL, and the runable program was also transferred to our computer. Research was begun on a multirate speech compression system capable of operating over a wide range of data rates.

SPEECH COMPRESSION AND SYNTHESIS

Quarterly Technical Progress Report No. 5
8 June 1979 - 7 September 1979

ARPA Order No. 3515

Name of Contractor:
Bolt Beranek and Newman Inc.

Effective Date of Contract:
6 April 1978

Contract Expiration Date:
7 July 1980

Contract No. F19628-78-C-0136

Principal Investigators:
Dr. John Makhoul
(617)491-1850 x 4332

Dr. R. Viswanathan
(617) 491-1850 x 4336

Sponsored by

Defense Advanced Research Projects Agency (DoD)
Monitored by RADC/ETC

The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the Defense Advanced Research Projects Agency or the U.S. Government.

TABLE OF CONTENTS

TABLE OF CONTENTS (CONTINUED)

1. SUMMARY

In this Quarterly Progress Report, we present our work performed during the period June 8, 1979 to September 7, 1979.

1.1 Introduction

During this past quarter we performed research in the areas of multirate speech compression, phoneme recognition, phoneme synthesis and we initiated the transfer to BBN of the M.I.T. text-to-speech system. Each of the following subsections refers to one of the sections of the QPR.

1.2 Phonetic Synthesis

Our main effort in the synthesis project this past quarter has been directed toward development of program modules to test the diphone templates. There has also been a substantial effort at labeling the short nonsense utterances in order to extract diphones from them. There was also some work in improving the algorithms for time-warping and smoothing the parameters in the diphone templates.

Section 2.1 describes the testing programs. Section 2.2 discusses the changes to the synthesis algorithms, and Section 2.3 reviews the status of the phonetic transcription (labeling) of the diphones.

1.3 Phonetic Recognition

This quarter we began work on a phonetic recognition program that will interface to the phonetic synthesizer to result in a very low rate phonetic vocoder. This phonetic recognizer will start with speech as input and will produce a sequence of phonemes, each with its own duration and pitch value. As discussed in the proposal we are following two approaches for the phonetic recognizer.

The first approach is to use a network of diphone templates to be matched against the input speech. The diphone templates used are similar to those used in the phonetic synthesizer, but are compiled into a structure suitable for matching. The diphone template approach to phonetic recognition requires a spectral distance metric to allow comparison of the diphone template spectra with the input spectra. In order to test different spectral distance measures we implemented a template-matching program that runs on the PDP11 using the FPS AP-120B for most of the computations. This program can also be used as an isolated word/phrase recognizer.

The second approach to phonetic recognition is Acoustic-Phonetic Recognition (APR). This method involves using acoustic parameters to try to determine phonetic features based on knowledge of the speech production process. At present, we have started to design how such a program should be implemented given our past experience with APR programs and knowledge gained concerning spectrogram reading in recent years. We have also made additions to the Acoustic-Phonetic Experiment Facility that will make the development of a better APR program feasible.

1.4 Text-to-Speech

This effort is aimed at transferring the MIT Text-to-Speech system to BBN. One of the major tasks is to convert the system from BCL to BCPL since only the latter is supported at BBN. Most of the modules have been converted successfully. There are still some library functions for which we do not have the source code, and there are some changes to be made. We will be aided in completion of this task by the MIT staff. We do have a runnable core image on line at BBN that can be used for testing.

1.5 Harmonic Deviations

As part of our effort to improve the naturalness of phonetic speech synthesis, we have started to investigate the addition of harmonic deviations to the LPC spectral representation used by the synthesizer. The harmonic deviations, which are extracted from the analyzed speech, are the difference between the amplitude of each harmonic of the pitch and the amplitude of the LPC model for the spectrum. In order to ascertain the feasibility of using harmonic deviations, we have added the harmonic deviations extraction and synthesis to our LPC vocoder. The speech resulting from this new vocoder is more similar to the original than that produced by the old LPC vocoder.

1.6 Multirate Coding

We completed the simulation of a 9.6 kb/s adaptive baseband transform coder. We also began the simulation of a fullband 16 kb/s adaptive transform coder. Both coders will serve as a basis for the multirate system we plan to test in the coming months. The multirate feature is achieved by stripping off bits from the output of the high rate system to operate in the range 2.4 - 9.6 kb/s. Our work to date is discussed in Section 6.

2. PHONETIC SYNTHESIS

Our main effort in the phonetic synthesis project this past quarter has been directed toward development of program modules to test the diphone templates. There has also been a substantial effort at labeling the short nonsense utterances in order to extract diphones from them. There was also some work in improving the algorithms for time-warping and smoothing the parameters in the diphone templates.

Section 2.1 describes the testing programs. Section 2.2 discusses the changes to the synthesis algorithms, and Section 2.3 reviews the status of the phonetic transcription (labeling) of the diphones.

2.1 Diphone Testing Programs

During the first year of the synthesis project, we found that when a synthesized sentence does not sound natural it is hard to find the cause. One reason is that it is very difficult to be sure of which phoneme contains the problem. Consequently, we designed and implemented a diphone test program which allows us to test a particular set of diphone templates. This program asks the user for a vowel and a vowel duration. It then creates a test sentence of the form:

for each of 23 consonants. For instance, for the vowel [IY] (as in "beet") and the consonant [M], the program synthesizes

M IY M - IY M IY - M IY M IY M IY M.

The user can then listen for problems in each of these 23 sentences, and check the transcriptions for the source of problems.

The test program uses a few simple rules to control the duration and pitch contours within the sentence in order that the synthesized test sentences sound natural.

Since we have started using this program to test the diphones that were labeled earlier, we have found several errors of various types. We have also begun to benefit from its use in terms of better intuitions for the most appropriate way to label other diphones.

To facilitate rapid testing of corrections to the labeling of a diphone, the diphone template compiler program (COMPOZ) has been changed so that incremental changes are possible without having to recompile the entire data base - a very lengthy process.

2.2 Synthesizer Changes

2.2.1 Gain Normalization

As has been mentioned in QPR-3 [1] and in the Annual Report, [2] each group of recorded diphone utterances has associated with it two normalization utterances. The synthesizer interpolates

Re

be
a
ha
of
fo
di
di
lo

no
sy
si

sy
ti
of
th
is
fo
wa
de
fa

:malization utterances to compute ›hone utterance. The synthesizer alization level to set the level ample, if the normalization level 3 dB lower than that of a second zer will add 3 dB to the first that the speaker was talking at a

g the diphones by synthesizing : few complete sentences that we oblems with inappropriate level perienced before.

ıe time warping algorithm in the ed in the various QPRs, the diphone template as being made up ic regions. Thus, the region of surrounding the phoneme boundary iddle region of the phoneme. The sly resulted in unreasonable time : was many times longer than the :he formula for the time warping ion has been changed.

Figure 1 shows the formula inelastic region as a function

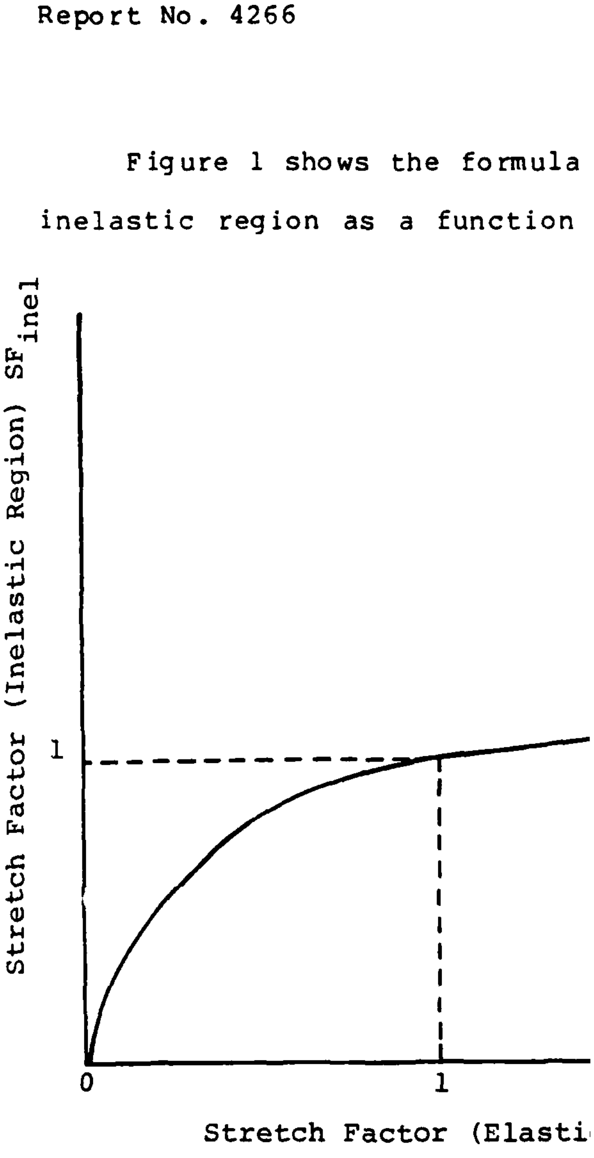


Fig. 1: New two-region fo warping

elastic region. The stretch multiplied by the template durat So if the stretch factor is 2 region is stretched to twice its in the figure, when the stretcl greater than 1 (i.e. the template must be stretched to match the required duration) the inelastic region stretch factor is closer to 1. The formula in this region is

$$SF_{inel} = SF_{el} \ (1-B) + B$$

where SF_{el} and SF_{inel} are the stretch factors in the elastic and inelastic regions respectively, and where B is a program variable with a value around 0.9. For example, if $SF_{el}=2$ and $B=0.9$, then $SF_{inel}=1.1$.

If the stretch factor is less than 1, then the formula is a quadratic:

$$SF_{inel} = -B \ SF_{el}^{2} + (1+B) \ SF_{el}$$

This particular quadratic is used because its derivative is the same as that in the previous formula at the boundary ($SF_{el}=1$).

The program is given the durations of the elastic and inelastic regions in each half of the template (each half of the template corresponds to half of one phoneme) and the required total duration for that half of the phoneme in the synthesized speech. It then solves for the stretch factors using the two formulae given above.

We have found that this new procedure has resulted in the elimination of some of the unnatural transitions that were present previously.

A second change to the time warping formula involved the effect of speaking rate on the "pronunciation" of the diphones. We had previously made the assumption that the effect of speaking rate on time-warping of each phoneme was symmetric about the middle of the phoneme. We have observed, however, another effect. In very slow speech there tends to be a relatively fast attack for each phoneme and a slower decay. Figure 2 shows a typical energy track for a vowel spoken at two different rates. As can be seen, the longer version decays more slowly (even after accounting for the overall duration change) than does the shorter version. Though not shown here, the spectral parameters also exhibit the same non-symmetric time warping characteristics.

The solution in the time warping algorithm has been to map the middle of a long phoneme (the ends of two diphone templates) to a point after the middle of the required phoneme. Thus, if Fig. 2a shows the energy track as reconstructed by concatenating two diphone templates, then when shortening this phoneme to the desired phoneme duration shown in 2b, the shape of the contour on the right side is changed by mapping the template onto the phoneme asymmetrically. This mapping (indicated by the dotted line connecting the two energy tracks in Fig. 2) should result in more appropriate pronunciation over a wider range of speaking rates. We have not yet fully tested this new feature.

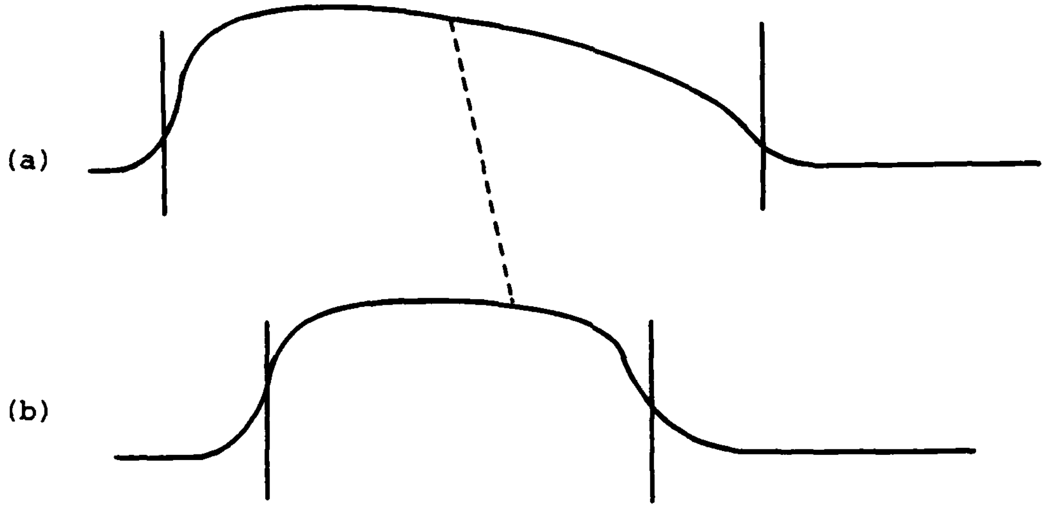


Fig. 2: Non-symmetric time warping. a) Shows a typical energy track for two halves of a long vowel phoneme reconstructed from two diphone templates. The dotted line indicates where the templates meet (corresponding to the middles of the phonemes in the nonsense utterances). b) Shows a typical energy track for a shorter vowel phoneme. The middle of the vowel in (a) is mapped (dotted line) to a point after the middle in (b) to produce the desired change in the shape of the contour.

2.3 Labeling

During the past quarter we have been testing old diphones and labeling new diphones in parallel. Currently, there are 1900 diphones labeled. Now that the testing procedure has been established, the labeling is expected to progress at a faster pace. By the end of the current quarter, there will be an instance of every diphone. All that will be required (after the current quarter) will be further tuning of the algorithm and the data base.

3. PHONETIC RECOGNITION

Since we intend to use the output of a phonetic recognizer to supply the input to the currently available diphone synthesizer we know that the recognizer must produce a sequence of phonemes each having its own duration and pitch. Our concern in the design of the two phonetic recognition schemes discussed below is how this information can be reliably obtained. Section 3.1 describes our diphone template approach to phonetic recognition. Section 3.2 discusses some of the spectral distance measures that might be used in this approach. In Section 3.3 we talk about a parallel effort at phonetic recognition using acoustic-phonetic features.

3.1 Diphone Template Recognition

A great amount of thought has gone into the design of a phonetic recognition system using diphone templates. What we hope to communicate here is the kinds of issues that were considered and indicate how their consideration has influenced the design. Although the design has not yet been completely finalized, it has stabilized recently to the point where implementation was started. In this section, it is assumed that the reader is familiar with the proposal [3].

The issue discussed here is that of using diphone templates and a matcher to do phonetic recognition. There are several factors that are relevant to this decision. First, we know that the output of this phonetic recognizer is going to be used in conjunction with a phonetic synthesizer that also uses diphone templates. Therefore, there is a strong motivation not only to use diphone templates in the recognizer but also to use the same set of diphone templates. That the same set of diphone templates should be used for both is motivated by the fact that since the recognizer is going to be used in conjunction with the synthesizer, using identical diphone templates for both will, at least, guarantee that the synthesized phonemes are spectrally close to the original.

3.1.1 Basic Method

The basic method of phonetic recognition based on the use of a network of diphone templates was described in our proposal [3]. Since the proposal, a few of the details have been changed.

The proposal referred to spectral templates in the finite state model (FSM) and indicated that scoring between the input speech and the spectral templates would be accomplished by classifying each input spectrum as one of a small number of spectral types, and then comparing input spectral types with template types by using an empirically derived confusion matrix. We have decided that defining the fixed set of spectral templates needed for this method might be a source of trouble and that the possibilities for the inclusion of new templates is made difficult because of this choice. Therefore the current representation does not use template labels but rather uses the spectrum from the template itself and a scoring metric in order to produce the scores needed for the matching process. This new representation scheme allows the possibility of a more ideal template scoring strategy based on the modeling of multi-dimensional probability distributions for spectra from a set of samples of that diphone. At the moment, the addressing capabilities of the PDP-1Ø (on which this work is being done) limit this more general approach.

3.1.2 Diphone Network Representation

The current design of the diphone network is quite similar to the example given in the proposal, however certain details have been specified in considerable more detail. In particular, the network is now composed of two different node types. The first node type corresponds to a transition in the Finite State Machine (FSM). Each such node consists of a complete spectral template (all 14 LAR coefficients and gain) as well as current statistical information regarding the "duration" of this template in its typical use (i.e., a probability density of the durations which have been observed thus far in training).

The second node type corresponds to a labeled state in the FSM and is used for the identification of completed diphone models and has neither spectral nor duration information. In the current design of the network, the only place where different paths can come together is at this second type of node.

Nodes are connected to each other by a directed path (and there is no provision within the network to follow a path in the reverse direction). Each node of the first type (where there is a spectral template) has an implicit self loop. The scoring used when this self loop path is taken depends completely on the duration information which has been collected during the course of previous training.

The network representation as it has been described so far would easily permit a matcher to determine diphone durations. However, since the synthesizer requires phoneme durations, we will explicitly mark each diphone path in the network at the point which corresponds to its phoneme boundary.

3.1.3 Diphone Network Compiler

An important consideration in the design of the diphone network compiler was that it be able to read diphone text files of exactly the same format as those read by COMPOZ (the program used to compile the parameters for the diphone templates into a single file which is used by the synthesizer program). The major reason for this requirement is that it will make it possible (once the diphone network compiler is written) to produce immediately a diphone network from our synthesis data base. Like COMPOZ this compiler also should permit the inclusion of incremental changes. That is, we would be able to replace diphone paths, or add additional optional paths to the existing binary file without rerunning the compiler on the original diphone templates. Not only does this preserve program compatibility and eliminate the necessity for redundant representation of the same information but it also greatly reduces the amount of time necessary to produce a new large network if it is only slightly different from a previously compiled one.

Another requirement on the format of the information compiled is that it must be such that statistical information can later be added by the matcher and be available for subsequent incremental additions.

Since the synthesizer can handle context constraints on any diphone template, we require that the recognizer also permit the definition of diphones in context (which, when recognized will reflect the appropriate context).

Preliminary estimates on the probable amount of information necessary to define a diphone network of this complexity strongly suggest that care be taken to produce a relatively compact network structure. This is important because if the network cannot be easily accessed (e.g. all of the network fits in the address space so that nodes can be directly pointed to) the execution time required by the diphone recognizer may be increased considerably.

3.1.4 Matcher

The design of the matcher has been strongly motivated by the following 4 considerations.

1) Sound Scoring Strategy
2) Continuous operation
3) Alignment availability for training
4) Efficiency

Of these four considerations the first is perhaps the most important. We feel that the scoring procedure should implement (as accurately as possible) a well formulated scoring strategy. The scoring philosophy requires that the score have components that are derived from a knowledge of the par'.icular path chosen through the network (a priori), the amount of speech aligned with each particular spectral template in the network (durations), and the score derived from a spectral comparison between the input spectrum and the template spectrum. It was the accurate evaluation of this last component (the spectral scores) which has influenced us to preserve the actual spectrum in the template model (rather than classifying each input and template spectrum by a label and scoring input labels against template labels with an empirically derived confusion matrix). Initially we will use a scoring metric to produce a score between the input spectrum and the spectral template. We have also considered the possibility of directly estimating the probability of the input spectrum using a suitably large collection of sample spectra. The biggest hindrance to this second step is the memory constraints that we have currently. This memory constraint may be eliminated in the near future by the addition of an extended addressing capability to the TOPS-20 monitor.

A second major consideration is that the matcher operate continuously, producing its output as it receives its input - with some delay - before acquiring all of the input. Thus the matcher can be thought of as producing output as soon as it has what it thinks is sufficient evidence to make a conclusion. Operated in this mode, further input, regardless of what it is, cannot cause the matcher to change previously produced output. This is important because of the intended use of the recognizer in a real-time vocoder application.

The third consideration, that of permitting alignment determination, is important for the continuing training of the recognition system, as well as to allow debugging of the recognizer.

The final consideration of efficiency has affected the design thus far in that the matcher will detect the merging of paths in the network and only remember the best of the merged paths. Furthermore, the matcher will be pruning the paths continuously so as not to waste computation time on improbable paths. We have designed the matcher so that the pruning can be based on either the number of theories or a score threshold, either of which is dynamically changeable.

Since the matcher will find (and remember) the best path through the network, it can output the best phoneme sequence and (with relatively little additional computation) their corresponding durations. Once a phoneme and its duration have been determined, the pitch will be calculated using a method similar to the weighted least squares method currently used to determine pitch values for our diphone synthesizer.

3.1.5 Training

During a training period the matcher would be run on large numbers of spoken sentences. Its alignment (forced to be correct if necessary) will be used to augment the statistical information distributed throughout the network. Since the matcher is designed to remember the alignment of its chosen best path through the diphone network the training can be continued on indefinitely, even when the matcher is in real use.

3.2 Spectral Distance Investigation

An integral part of the network diphone recognition program is the spectral distance measure used to score a match between unknown

Re

sp
re

Ho
re
of

In

av
sp
in

We

Ar
a

Lo

W
s
I

mplates. The performance of this
as good as the distance measure.

ral Error Measures

measures is, of course, infinite.
es have been proposed for speech
n purposes. Below we mention some
uss each briefly.

on

for spectral quantization is the
in the log area ratios between two
uantizing each of the parameters

og Area Ratios

are error between two sets of Log
variable-frame-rate transmission
.

easure)

sed in isolated word recognizers.
ric verification component of our
h a dynamic programming algorithm.
basic distance metric in the HARPY

network speech recognition algo
spectra by looking at the en
filtering one LPC filter w
deficiency in the algorithm
overall energy between the two
all frequency bands equally.
measure is not symmetric.

Mel-Scale Cepstral Coefficient

This metric warps the fr
scale warping, which is in
dependence of the human ear. S
metric to be more effective
recognizing isolated words.

Other Metrics

One could easily devise
measures to be used for spee
would propose to use a measure
is sensitive to the direction
a small number of metrics to do

3.2.2 Program to Test Spec

If we were to test diffe
the diphone recognizer, the ef

be overshadowed by the other characteristics of the matcher. Therefore, we have written a program that is a test bed for different spectral error measures. This program compares a set of test words with a set of reference words, and decides on the closest match for each test word. This is basically the same function as that of an isolated word matcher. The percentage of the time that the reference word chosen is another instance of the test word is a good measure of the efficiency of the spectral distance measure used.

It is important to remember that a spectral distance measure used for speech recognition must do more than just detect any audible differences between two spectra (as would be needed for an LPC vocoder). Rather, the measure should ignore - to some extent - audible differences that are not likely to imply a different phoneme - or a different speaker. For this reason, a good measure compares sequences of spectra rather than just single spectra. One of the metrics we have designed may be effective at looking at the formant motion as a whole.

3.2.3 Isolated Phrase Recognizer

As mentioned above, the error measure testing program can also be used as an isolated word and phrase recognizer. This program is written on the PDP11 using the FPS AP-120B to do the LPC analysis, the dynamic programming time warping, and the spectral distance calculation. Currently, the program can compare a 1-sec word to 20 1-sec reference word templates in 3 seconds. Changes in the I/O will speed this up to approximately 1 second.

The program has been tested with a set of 20 names of states. When the program was given a second set of 20 names spoken by the same speaker it correctly recognized all 20 of the names. The metric used was one of the simplest - the Euclidean Distance of Log Area Ratios. The program has not yet gone through extensive testing, and so this performance may not be representative.

3.3 Acoustic-Phonetic Recognition

In addition to the diphone network recognition approach to phonetic recognition, we will also be investigating the use of an Acoustic-Phonetic Recognition program. During the last years of the ARPA Speech Understanding Project, and since its suspension, we have accumulated a large amount of knowledge about the acoustic-phonetic rules that apply to speech. The results of experiments in human spectrogram reading [5] attest to this greater level of knowledge.

During the past quarter, we have begun to consider ways to capture some of the knowledge employed by the human spectrogram reader in the form of a computer program. The hope is that these procedures will complement the diphone network recognizer and improve its performance. The acoustic-phonetic recognizer will have the advantage (over the diphone recognizer) of being able to use specific knowledge pertaining to a specific phonetic distinction - rather than be restricted to a uniform metric.

As a preliminary effort toward this goal, the Acoustic-Phonetic Experiment Facility (APEF) [6] has been expanded with additional capabilities that make it possible to design new acoustic parameters and use them without having to write new programs for analyzing the speech data base. Thus, it now combines the capabilities of several other programs, while further simplifying the whole acoustic-phonetic research process.

4. TEXT-TO-SPEECH

As proposed, we obtained the M.I.T. text-to-speech system for immediate use in conjunction with our diphone synthesizer. Initially we only had the executable files of that system but were able to produce appropriate input for our diphone synthesizer (Phoneme sequence with corresponding duration and pitch values). Because the interaction with the text-to-speech programs was not at all tailored to the kind of use which we had in mind for it, we decided to modify the source code. The source code, however, was written in BCL for which we had no compiler. Furthermore, no one here at BBN was familiar with BCL. However, because BCL is a variant of BCPL (which we use), it was thought that converting the source code to BCPL for subsequent modification would be the most reasonable and convenient thing to do. We hired an M.I.T. student who had worked with the text-to-speech system (and was, therefore, familiar with BCL as a language as well) to do the language conversion for us. The work on the first phase of conversion took one month, at the end of which almost all of the 200 plus programs had been translated and could be successfully compiled with our BCPL compiler. Unfortunately we ran into a couple of unanticipated problems which have so far prevented us from taking advantage of this translation work in the way we had planned to originally.

First, we found that, in addition to the text-to-speech programs, a large number of system library functions would also have to be translated. What made this problem particularly serious was the fact that the source code for some of these programs could not be found.

Second, during the process of compiling the translated programs, compiler messages indicated problems that had existed in the original BCL code but which had not been detected by the BCL compiler at M.I.T. We then had to determine what the author of the programs intended in each such case. These errors, plus the fact that some of the BCL code had been changed at M.I.T. since we had started the translation, made the the prospect of compatibility less hopeful than we would have desired.

From the time all of this happened in the beginning of the summer, when the M.I.T. staff was not available for consultation no further translation effort was expended until just recently. In mid-September, Sheri Hunnicutt, an M.I.T. staff member who had worked on the text-to-speech project, wanted to clean up their system for outside distribution and so offered to help rectify the difficulties. This process has just begun and is proceeding at a rate determined by M.I.T. Extra translation effort on our part should be minimal.

5. HARMONIC DEVIATIONS

As outlined in the proposal, we started to investigate the feasibility of improving the spectral representation for LPC synthesis by the inclusion of the deviation of each spectral harmonic from the smooth LPC spectral model. In order to test out the principle of harmonic deviations we used it in an analysis-synthesis (vocoder) environment, to see the improvement in the quality of the speech. Below, we give a discussion of some of the possible implementations of this idea in a vocoder.

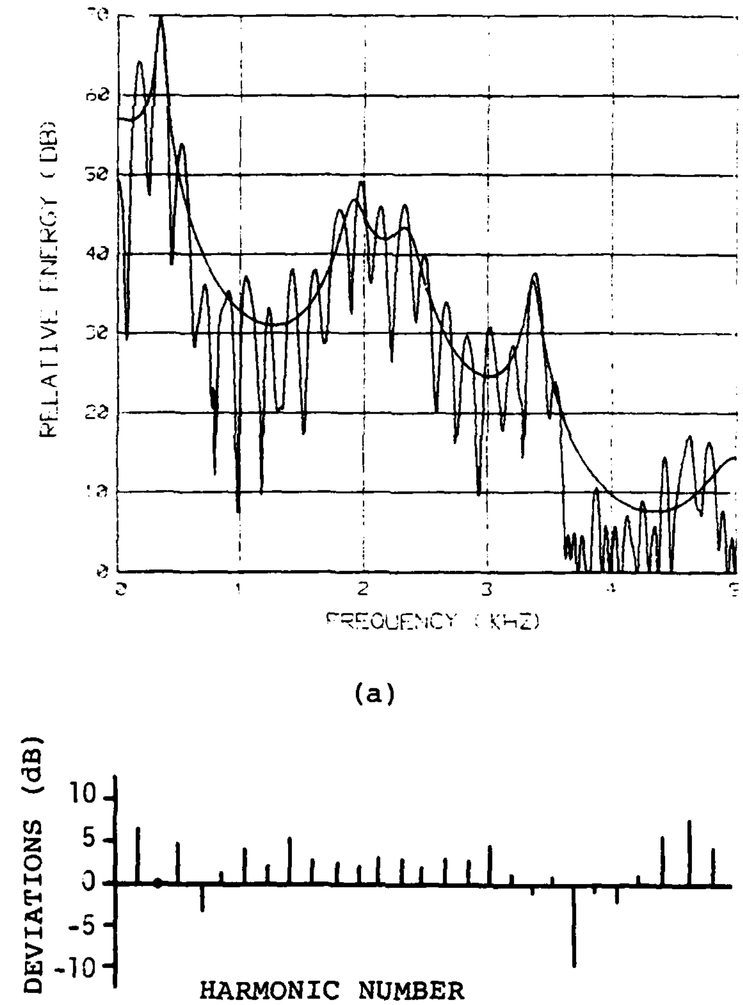
5.1 Harmonic Deviations Vocoder

The basic idea used in the Harmonic Deviations Vocoder (HDV) is as follows. The transmitter extracts from the speech signal and sends to the receiver the parameters of a smooth speech spectral envelope model as well as the deviation at each harmonic frequency between the speech spectrum and the spectral envelope model. The receiver generates a pitch-period of the excitation signal from a fixed frequency-dependent harmonic phase spectrum and the harmonic amplitude spectrum computed from the transmitted deviations. The excitation signal is in turn applied to the filter corresponding to the spectral envelope model to produce the output speech. Below, we provide a more detailed discussion of the HDV vocoder.

Figures 3 and 4 illustrate this procedure. Figure 3a and 4a show (for a male and female speaker respectively) a power spectrum and a smooth LPC spectrum superimposed. Figures 3b and 4b show the deviation between the two spectra expressed in dB. The spectral envelope may be obtained from standard LPC analysis of speech. Since the all-pole spectrum, $\hat{P}(\omega)$ resulting from the LPC analysis provides a good approximation to the envelope of the speech spectrum, the harmonic deviations are expected to be small relative to the absolute amplitudes and thus will require only a few levels to code them.

5.2 Extraction of Harmonic Deviations

The speech signal is analyzed at a fixed rate, say, once every 10 ms. The chosen frame of speech is Hamming-windowed and is subjected separately to 1) standard autocorrelation LPC analysis, and 2) power spectrum $[P(\omega)]$ computation via the FFT algorithm. The order of the FFT is chosen to be high, for example, to give a frequency spacing of about 10-15 Hz. The power spectrum $\hat{P}(\omega)$ of the LPC model is computed via the same order FFT from the finite impulse response of the linear prediction inverse filter A(z). The two power spectra are then expressed in dB. Next, all pitch harmonics are located by "peak-picking" on the signal power spectrum $P(\omega)$. For each integer multiple of the current quantized pitch frequency, say, $m\omega_0$, we consider the two harmonic peaks of $P(\omega)$ already available, one on each side of $m\omega_0$; at these two peaks, the dB differences between $P(\omega)$ and $\hat{P}(\omega)$ are computed, and a linearly weighted sum of the two deviations is calculated. This



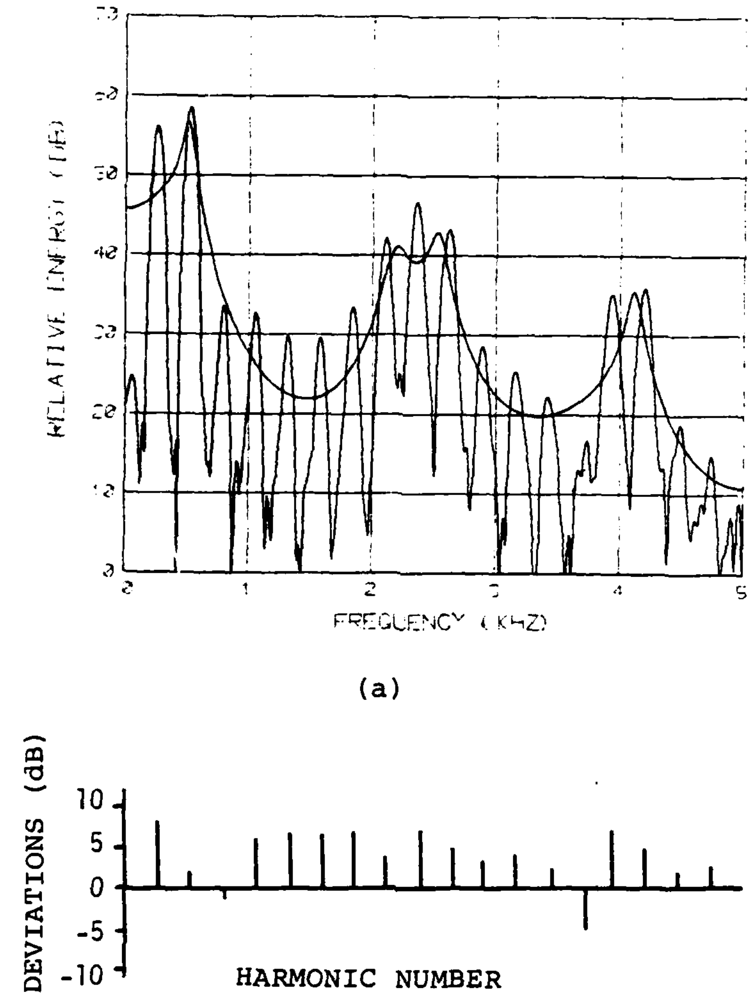
(a)

(b)

Fig. 3 Spectral errors at the harmonics of the fundamental F_0 in LPC modeling, obtained from a male voice (F_0 =180 Hz).

(a) Plots of the spectrum of the speech signal (ragged plot) and the spectrum of the corresponding 12-pole LPC filter (smooth plot).

(b) Deviations in dB between the two spectra in (a) plotted as a function of the harmonic number.



(a)

(b)

Fig. 4 Spectral errors at the harmonics of the fundamental F_0 in LPC modeling, obtained from a female voice ($F_0 = 280$ Hz).

(a) Plots of the spectrum of the speech signal (ragged plot) and the spectrum of the corresponding 12-pole LPC filter (smooth plot).

(b) Deviations in dB between the two spectra in (a) plotted as a function of the harmonic number.

last quantity is then assigned as the harmonic deviation at the frequency $m\omega_{\emptyset}$. After all the harmonic deviations are extracted, their mean or DC value is removed to obtain zero-mean deviations, which are then coded and transmitted.

5.3 Coding of Harmonic Deviations

Although our particular application to speech synthesis does not impose strict limits on the amount of storage, we consider here briefly the coding issues that would arise in a vocoder application. Given a fixed number of bits, the problem is: how to distribute the bits efficiently among all the harmonic deviations. One strategy is to transmit a fixed subset of the deviations (say the first 10 deviations corresponding to low frequencies). A more complex scheme is to use the LPC spectral envelope to determine the number of bits used for each deviation (See **Section** 6.3.1). In Fig. 5, we show a histogram of all the harmonic deviations, which was obtained using a quantization step size of 1.5 dB.

5.4 Synthesis

To synthesize speech we use the inverse DFT to compute one pitch period of the excitation signal, which is then applied to the all-pole LPC filter. The amplitude excitation spectrum is obtained directly from the harmonic deviations, while the phase spectrum given by the fixed group delay characteristic is determined by averaging several frames of voiced speech. The inverse DFT of the

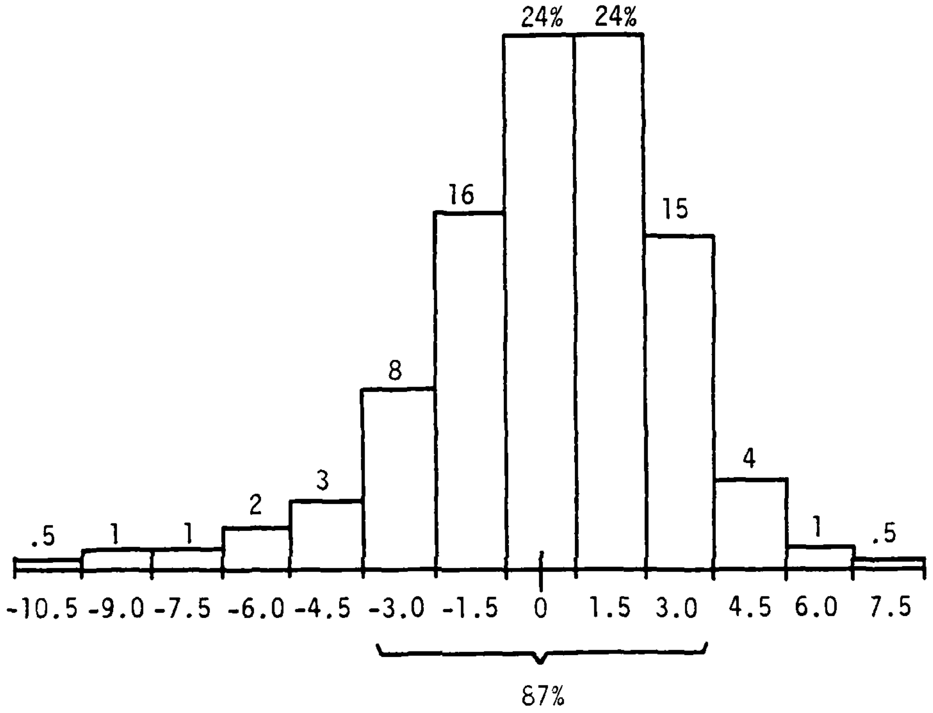


Fig. 5 Histogram of harmonic deviations obtained using a step size of 1.5 dB. The probabilities were averaged over all the harmonics, from a set of six utterances of male speech. The speech was sampled at 10 kHz and the spectral envelope was obtained using a 12-pole LPC analysis.

Fourier transform coefficients corresponding to the amplitude and phase spectra results in a pulse-like excitation signal whose power spectrum contains the corrections due to the harmonic deviations. The energy in the excitation signal is adjusted to be equal to G^2 by multiplying the samples of the excitation signal with an appropriate scale factor. Note that for the autocorrelation method

$$G^2 = R(0) \sum_{i-1}^{p} [1-K^2(i)], \qquad (6)$$

where R(0) is the transmitted speech-signal energy and K(i) are the reflection coefficients. When this scaled excitation signal is applied to the all-pole filter, the filter imparts the spectral envelope so that the output speech has the right amplitudes for those harmonics for which deviations have been transmitted.

5.5 Preliminary System

As part of our phonetic synthesis project we have developed a floating-point non-real-time simulation of an analysis-synthesis system that uses harmonic deviations. In this simulation, we do not quantize LPC parameters, and we employ the first 15 harmonic deviations. We extract the harmonic deviations from the 10-kHz sampled speech once every 10 ms. Our preliminary experiments have shown that the addition of harmonic deviations to the LPC synthesis significantly increases the naturalness and fullness of the synthesized speech while reducing the buzzy quality.

Ir

Repo

6.

6.1

test

code

mult

mult

of t

deta

allo

we g

disc

inve

6.2

(b).

gain

tran

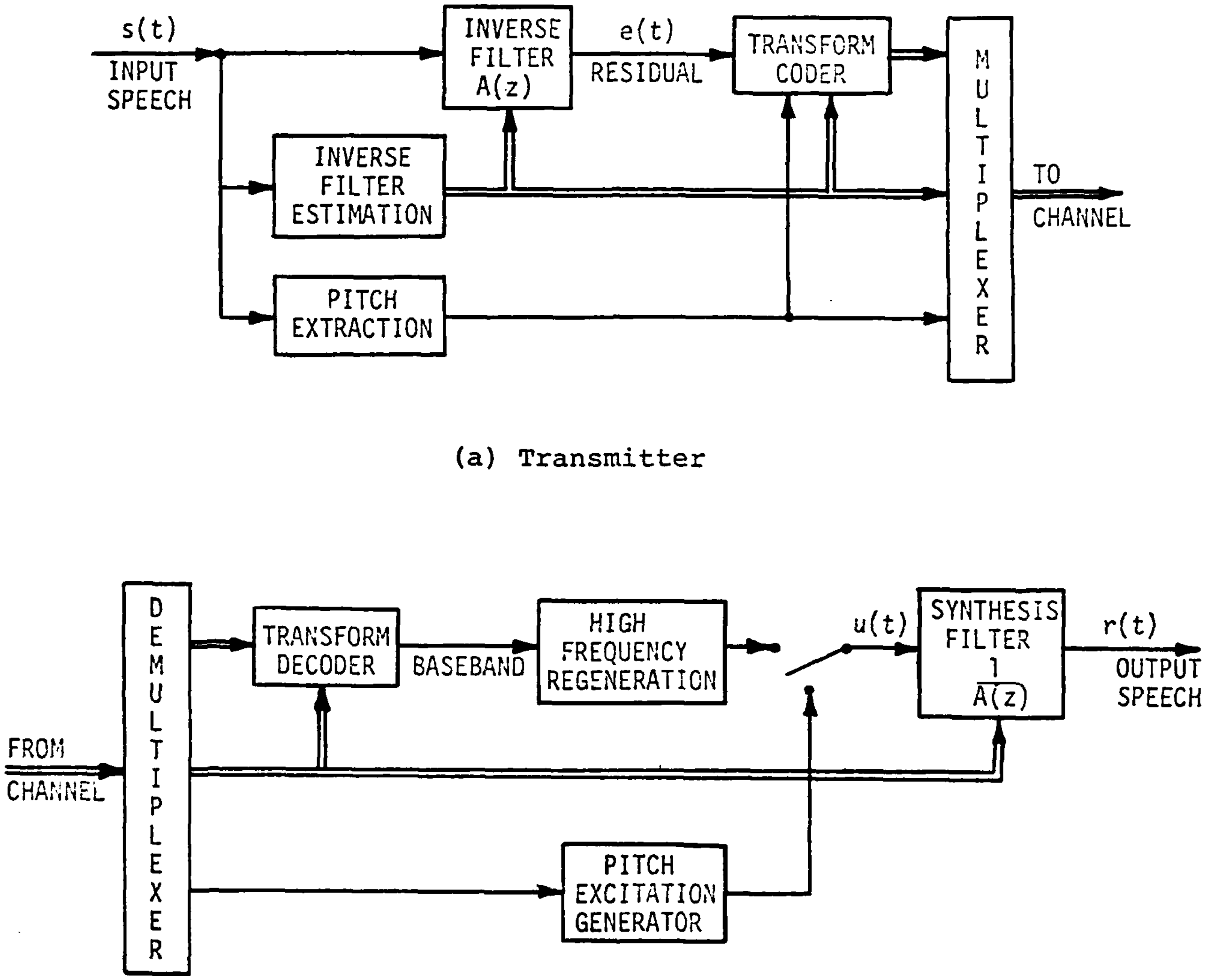
inve

obta

of t

ATC

the



(a) Transmitter

(b) Receiver

Fig. 6: Block diagram of proposed multi-rate system. The multi-rate aspect is achieved by controlling the number of transform coefficients transmitted. At the receiver, residual excitation is used at higher data rates, and pitch excitation at lower data rates.

low frequency portion of the DCT is transmitted, using a fixed number of bits per frame. The transmitted low frequencies constitute the baseband. The width of the baseband is fixed in time and corresponds to about 1.1 kHz, i.e., one third the original bandwidth.

At the receiver, the DCT of the baseband is decoded and the missing high-frequency components are regenerated. The method of high-frequency regeneration (HFR) was described in a previous report [2]. Once the fullband DCT is restored, an inverse DCT yields the received time-domain residual waveform. The latter waveform is used as input to the LPC all-pole synthesis filter to generate the output speech. We now discuss specific aspects of the above described baseband coder.

6.3 Details of the Algorithm

A marked difference between our transform coding approach and conventional ATC is that we code and transmit the DCT of the residual rather than the speech signal. In the proposal for this work [3] and in a recently published paper [7], we have shown that quantizing the DCT of the residual with an appropriately varying step-size, affords the same increase in signal-to-noise ratio over APCM as that afforded by ATC of the speech. The bit allocation scheme that we are using remains the same as that used in conventional ATC. We explain this aspect of the coder next.

6.3.1 Bit Allocation

In ATC of speech, the available bits to be used at each frame are judiciously distributed among the DCT coefficients. The allocation of the bits, at each frequency f, is done according to the following formula:

$$L(f) = E/|H(f)| \qquad (7)$$

where L(f) is the number of quantization levels used at frequency f, E is a proportionality constant, and $1/|H(f)|$ is the magnitude of the model of the speech spectrum. Taking the base 2 logarithm of both sides of (1), we obtain

$$b(f) = \log_2[1/|H(f)|] + b_\emptyset \qquad (8)$$

where b(f) is the number of bits to be used at each frequency f, and $b_\emptyset = \log_2 E$ is the average number of bits per DCT coefficient. In practice, when using a fixed-length binary coder, all b(f) are rounded off to integers, $\hat{b}(f)$, such that (i) no negative bit-assignment results, and (ii) the sum of all integer $\hat{b}$ is equal to the total available bits per frame. To satisfy the above two constraints, $\hat{b}(f)$ is derived iteratively from

$$\hat{b}(f) = \max\{\emptyset,[b(f) + \delta]\} \qquad (9)$$

where [.] denotes "integer part of". In (9), δ is changed at each iteration, until the final sum of allocated bits equals the number of available bits. At each frequency, each DCT coefficient is quantized using $\hat{L} = 2^{\hat{b}(f)}$ levels and it is coded with $\hat{b}(f)$ bits.

The spectral model used in (7) and (8) is made up of two components. The first component is $1/|A(f)|$, the magnitude of the LPC envelope of the speech spectrum with A(f) given by

$$A(z) = 1 + \Sigma\ a_k z^{-k} \qquad (1\emptyset)$$

where $\{a_k,\ 1\leq k\leq p\}$ are the predictor coefficients. The second component of $1/|H(f)|$ is $1/|P(f)|$, a model for the harmonic structure of the speech spectrum. In our experiments to date, we modelled pitch by means of the one-tap pitch filter

$$C(z) = 1 - cz^{-M} \qquad (11)$$

where c is the coefficient of the filter at the Mth lag, and M is the number of samples in a pitch period. M can be derived from the pitch value transmitted by the narrowband LPC vocoder. The model for the harmonic structure of the speech spectrum can be derived from C(z) in two ways. In the direct manner, we let

$$|P(f)| = |C(e^{j2\pi f})|$$

In the indirect manner, we obtain first the impulse response of the all-pole pitch filter 1/C(z). Second, we truncate this impulse response, by multiplying the samples by the analysis window (19.2ms). Third, we obtain the spectral model $1/|P(f)|$ by taking the magnitude spectrum of the truncated impulse response. We have experimented briefly with the two methods and found that the direct method yields better results.

In conventional ATC without spectral noise shaping, the quantization step-size is theoretically the same for all DCT coefficients, which should result in a flat quantization noise spectrum, i.e., maximum signal-to-noise ratio. However, in our implementation, the step-size is varied in frequency for three fundamental reasons. These are: (i) to allow for shaping of the noise spectrum because a flat noise spectrum is not optimal from a perceptual point of view, (ii) to account for the fact that the DCT coefficients to be quantized are those of the residual waveform instead of the speech, and (iii) to compensate for the rounding off of the values of b(f) to integers. Item (i) requires altering the basic bit allocation scheme and this is explained next.

6.3.2 Noise Shaping

From past experiments, we have determined that a perceptually preferred shape for the noise spectrum is not flat, but instead can be derived from the spectral envelope of the speech. In particular, we are currently using the form $1/|A(f)|^{\alpha}$ to determine the spectral envelope of the noise, with $0 \leq \alpha \leq 1$. The inverse of the desired spectral noise shape is used in the bit allocation scheme as a weighting function, such that the number of levels is given by

$$L(f) = E|A(f)|^{\alpha}/|A(f)||P(f)| \qquad (12)$$

where we have explicitly separated the original spectral model into its two components: LPC and pitch. Equivalently, the number of bits is given by

$$b(f) = (1-\alpha)\log_2[1/|A(f)|] + \log_2[1/|P(f)|] + b_0 \qquad (13)$$

where α is a parameter which we can vary in the range [0,1] to control the degree of spectral noise shaping. The case $\alpha=0$ isthe conventional bit allocation scheme, without any noise shaping, as depicted in (7) and (8). For $\alpha=1$, the noise spectral envelope is the same as the LPC spectral envelope of speech. For this value of α, it can be seen from (12) and (13) that the bit allocation is based on the pitch model alone. It can be shown that this case of noise shaping results in a small increase in signal-to-noise ratio over APCM due only to the pitch model. We believe that intermediate values of α will yield best perceptual results.

We note here that the effect of the weighting function in (12) and (13) is to increase the number of allocated bits at some frequencies while decreasing it at others, thus trading bits between the different regions of the spectrum. This constitutes a redistribution of bits relative to the original ATC case. Along with this redistribution of bits, we change the quantization step-size in frequency in order to accommodate the change in the available levels. It is this change in step-size that effectively shapes the quantization noise spectrum, and we explain it in the next subsection.

6.3.3 Step-Size Control

As mentioned earlier, the quantization step-size in conventional ATC is fixed in frequency. We have shown elsewhere [3,7] that since we wish to quantize the DCT coefficients of the residual, the quantization step-size should be made proportional to the magnitude spectrum of the LPC inverse filter, in order to achieve the same gain in signal-to-noise ratio over APCM as that afforded by ATC of the speech. Thus, the quantization step-size is given by

$$D(f) = D_{\emptyset}|A(f)| \tag{14}$$

where $D_{\emptyset}$ is a proportionality constant to be determined experimentally. It is important to note here that (14) describes the <u>relative</u> change in step-size from one frequency bin to the next. At each frequency, a uniform $\hat{L}(f)$-level quantizer is used with step-size D(f). In this manner, the quantization noise variance varies in frequency according to (14). $D_{\emptyset}$ in (14) can be thought of as an overload factor. It is derived experimentally to achieve a suitable tradeoff between granular noise and clipping (overload) distortion.

In addition to the step-size variations given in (14) we impose another variation on D(f) to insure proper noise shaping. With noise shaping, (14) becomes

$$D(f) = D_{\emptyset}|A(f)|/|A(f)|^{\alpha} \tag{15}$$

where we have explicitly shown that the step-size varies in proportion to the desired spectral shape of the noise $1/|A(f)|^{\alpha}$.

Finally, we vary D(f) in a manner that takes into account the fact that the number of levels used is a power of 2. The actual number of levels used is $\hat{L}(f)$ and is different from the ideal number of levels, L(f), which would have resulted from the bit-allocation scheme, were it not for the integer-bit constraint. In general, L(f) and $\hat{L}(f)$ can be related to one another by

$$L(f) = a(f)\hat{L}(f) = a(f)2^{b(f)}$$

where a(f) is a number in the range $\emptyset.5<a(f)<2$. The value of a(f) is obtained during the iterative bit allocation scheme. To compensate for the change in the number of levels, from ideal to actual, we change the step-size accordingly. Thus, from (15), the final overall control of the step size becomes

$$D(f) = D_{\emptyset}|A(f)|^{(1-\alpha)}a(f) \tag{16}$$

We note here that in (16) we do not use the pitch model to control the step-size. The reason is that the pitch filter is not used to inverse filter the speech, nor is it used in shaping the noise.

6.4 Preliminary Results

In our preliminary investigations of the baseband transform coder we have used input speech signals bandlimited to 3.33 kHz and sampled at 6.67 kHz. The frame size is 128 samples, i.e., 19.2 ms, the same as that used in the LPC pitch-excited vocoder. We use an 8-pole LPC model for the speech spectrum. In addition to the 8 log-area-ratios, we code and transmit one gain value, a pitch value, a pitch coefficient, and two HFR codes. The bit-rate for this side information is 2.75 kb/s. For a total of 9.6 kb/s, there *remains about 130 bits per frame to be used for coding the* DCT. The width of the baseband is fixed at 1.1 kHz, i.e., only the first 42 DCT components are coded and transmitted.

We experimented with the two forms of the pitch spectral model and found that the use of the spectrum of the pitch inverse filter C(z) is superior to the use of the spectrum of the truncated impulse response of the pitch filter 1/C(z).

We used three values for the noise spectral shaping parameter in (6): α=0.0, 0.5, and 1.0. We found the results from 10 sentences to be perceptually similar for the three cases considered. It is difficult to distinguish between the three cases because the output speech quality is degraded by the combined effects of HFR and baseband quantization noise.

In general, the performance of the coder for male voices was preferred over its performance for female voices.

6.5 Future Research

The results presented above are preliminary in nature and more research is needed to improve the quality of the speech obtained with the baseband transform coder. We feel that further investigations are needed and will help set the stage for the multirate coding systems we are about to test.

As for the design of the multirate coder, the basic approach we plan to investigate is to start with a coder designed to operate optimally at some fixed bit rate. For example, such a coder could be the above described 9.6 kb/s baseband coder or it could be a fullband ATC scheme at 16 kb/s.

There are two basic methods to let the channel strip off bits to achieve data rates below 9.6 kb/s. In the first method, the bits, i.e., the codes representing the DCT components, are ordered from low to high frequencies. When the channel discards bits the high-frequency components are dropped out first. Whether we start with a baseband 9.6 kb/s coder or with the fullband 16 kb/s coder, the effect of the missing codes will be the same, namely, a lack of high-frequency components. The receiver regenerates the missing components to reconstitute the fullband residual prior to synthesis.

In the second method, the codes of each frame are ordered in such a manner that the bits occur in consecutive sequences, with the first sequence being the most significant bit of every code, the second sequence the next most significant bit of the codes, and so on, until the last sequence, which is made of the least significant bit of every code. When bits are discarded, the least significant bits are dropped out first. This action corresponds to having coarser quantization on the DCT components. For example, if each code is decreased by one bit, this corresponds to doubling the quantization step-size (or dividing the number of levels by 2). In addition, this method will cause the suppression of certain frequency components, namely those that are coded using short codes (e.g. 1 and 2 bits). The absence of certain DCT components will result in spectral gaps to be filled by the receiver in a manner similar to HFR.

We plan to test the above discussed approaches to multirate coding during the coming months.

7. REFERENCES

[1] Makhoul J., et. al., "Speech Compression & Synthesis," Quarterly Progress Report prepared for Advanced Research Projects Agency, BBN Report No. 4Ø61, 6 Oct. 1978 to 5 Jan. 1979.

[2] Berouti, M., et.al., "Speech Compression and Synthesis," Annual Report prepared for Advanced Research Projects Agency, BBN Report No. 4159, August 1979.

[3] Makhoul, J., et. al., "Proposal for Continuation of Research on Narrowband Communication," Submitted to Advanced Research Projects Agency, BBN Proposal No. P79-ISD-2Ø, November 1978.

[4] Davis, S., "Order Dependence in Templates for Monosyllabic Word Identification," IEEE Int. Conf. on Acoust. Speech & Sig. Proc., April 1979, pp. 57Ø-573.

[5] Zue, V. and Cole, R., "Experiments on Spectrogram Reading," IEEE Int. Conf. on Acoust. Speech & Signal Proc., April 1979, pp. 116-119.

[6] Schwartz, R., "Acoustic-Phonetic Experiment Facility for the Study of Continuous Speech," IEEE Int. Conf. on Acoust. Speech, & Signal Processing, April 1976, pp. 1-4.

[7] Berouti, M. and Makhoul, J., "An Adaptive-Transform Baseband Coder," Proceeding of the 97th Meeting of the Acoustical Society of America, paper MM1Ø,June 1979, pp. 377-38Ø.

END

FILMED

7-85

DTIC

the